In the claims

Cancel claims 8, 16, 23, 29, 33, 35, and 50 without prejudice or disclaimer.

Add the following new claims

57. The system of claim 1 wherein the computer readable components are associated with an application and are stored on a computer readable medium.

58. A system for maintaining security in a distributed computing environment, comprising:

a policy manager for managing a security policy; and

an application guard for managing access to a transaction related with an application as specified by the security policy.

59. A system to protect computer systems against unauthorized access for managing and enforcing complex security requirements in a distributed computer network comprising:

a policy manager located on a server for managing and distributing a policy to a client; and

an application guard located on the client, acting to grant or deny access to various components of the client, as specified by the policy.

60. The system of claim 59 wherein the server is connected via a network to the client.

61. The system of claim 59 wherein the server is connected to many clients.

62. The system of claim 59 the server further comprising:

a central processing unit;

a Read-Only Memory (ROM);

a Random-Access Memory (RAM);

a non-volatile memory;

an input device; and

a display;

wherein the ROM, the RAM, non-volatile memory, input device and display are connected via a bus.

63. The system of claim 59 wherein the policy manager is a program located on a server in non-volatile memory.

64. The system of claim 59 wherein the client contains a program stored in non-volatile memory for granting or denying access to various components or resources of the client, as specified by the policy distributed from the server.

65. The system of claim 59 wherein the server includes a non-volatile memory, where the policy manager is located that specifies the security requirements for applications and database objects;

said policy contains security rules that describe at least one constraint that constrains which applications a particular user can access and

which objects within an application a user can access.

66. The system of claim 65 wherein the policy manager allows the administrator to choose whether the constraints are effected by any of

time,

geography, and

external events.

67. The system of claim 59 wherein the policy is capable of constraining access to applications and operations within applications.

68. The system of claim 59 wherein the policy is organized into groups and hierarchies.

69. The system of claim 59 wherein the policy includes access rules, which include:

a grant rule that grants a privilege to a subject on an object under a first constraint; and

a deny rule that denies a privilege to a subject on an object under a second constraint.

70. The system of claim 59 the policy manager further comprises

a management station program to operate the policy manager,

a distributor program to distribute local client policies to clients, a logger program to track authorization requests, and

a database management system (DBMS) to maintain policy data files.

71. The system of claim 59 wherein the policy manager further comprises:

an audit log data file to record authorization requests;

an optimized policy data file;

an enterprise policy data file;

an administrative policy data file; and

a local administrative policy data file.

72. The system of claim 59 wherein the policy manager is located within non-volatile memory.

73. The system of claim 59 wherein the policy manager allows system users to

implement,

analyze,

edit and

update

a centrally-managed policy.

74. The system of claim 59 wherein the policy includes at least one policy rule comprising:

1) an object that is to be protected;

2) an access right or privilege;

3) a global or local user to which the privilege applies; and

4) conditions under which the privilege is granted or denied, wherein the user is given a choice of types of conditions including

whether to use built-in access criteria wherein the user can select

whether to use time of day and

whether to use location,

and

whether to use custom-defined access criteria.

75. The system of claim 59 wherein the policy manager comprises a Graphical User Interface (GUI) that provides at lest a user-friendly set of menu options or management services to fully operate the policy manager and control programs that perform at least

navigation,

searching,

distribution,

editing,

querying, and

log viewing.

76. The system of claim 59 wherein the policy manager comprises an Application Programming Interface (API) that at least allows programs to perform the same functions as a human operator.

77. A system comprising a computer having a security policy that includes at least one or more components having at least:

an object,

a subject,

a privilege, and

a condition.

78. The system of claim 77 wherein each object can be an application or an operation within an application.

79. The system of claim 77 wherein the object is capable of being set to be at least any of:

an application,

a method,

a web page,

a database table

a file, and

one or more menu items in a graphical user interface.

80. The system of claim 77 wherein the object can be organized into at least an object hierarchy such that:

if a user is granted a certain privilege on a parent object, then that user is automatically granted the privilege on all the children objects, and if the user is denied a certain privilege on a parent object, then that user is denied the privilege on all the children objects.

81. The system of claim 77 wherein the privilege is capable of being inherited from a parent to a child object.

82. The system of claim 77 wherein the subject is capable of being set to be at least any of:

a user and

a role containing one or more users,

who can at least

access a protected object, and

have access to at least some information in the system.

83. The system of claim 77 wherein the subject is capable of being a user that can be chosen to be either internal or external to a system.

84. The system of claim 77 wherein the object comprises a list that is capable of containing one or more users authorized to access the object who can log on to the object and be authenticated by the object through an external authentication server.

85. The system of claim 77 wherein the system is capable of having the subject be a user who at least:

can be maintained separately by one or more components each of which can be an object or directory server; and

can be extracted from said one or more components to synchronize the components thereby maintaining their access current.

86. The system of claim 77 wherein components comprise an alias-user who at least inherits all privileges of a user under certain conditions, thereby facilitating authorization management by providing fine granularity of control on propagation of a privilege.

87. The system of claim 86 wherein the system is capable of having the alias-user be created to perform a job function while the user is absent, and inheritance of the privilege takes effect only when the user is absent.

88. The system of claim 77 wherein a user of the object is capable of being defined to be local to the object.

89. The system of claim 77 wherein a user can be at least a global user mapped to a set of local users having at least one local user per object.

90. The system of claim 77 wherein the privilege defines at least one kind of access that is allowed to the object and includes at least one right to perform a particular action on the object.

91. The system of claim 77 wherein the privilege is capable of including at least

a right to execute an application,

a right to download a web page,

a right to query a database table, or

a right to view a menu item.

92. The system of claim 77 wherein the component is capable of being assigned to be a wild card that is capable of being used at least as a privilege, object, or subject.

93. The system of claim 77 further comprising an access request that includes at least:

a privilege,

an object, and

a subject;

wherein the access request is used by at least a subject to request authorization of at least a privilege on at least an object.

94. The system of claim 93 wherein the access request at least:

matches a grant rule if the privilege, object, and subject match those in the rule, and the constraint in the rule is met; and

matches a deny rule if the privilege, object, and subject match those in the rule, and the constraint in the rule is not met.

95. The system of claim 77 wherein an access request is at least:

denied if

there is a deny rule matching a request, or there are no access rules matching the request; and

granted if there are no deny rules matching the request and there is a grant rule matching the request.

96. The system of claim 77 wherein conditions comprise constraints and the system having at least facilities for defining constraints as expressions formed from operators including at least NOT, AND, and OR.

97. The system of claim 77 wherein conditions at least:

are constraints on when the object or the subject can be accessed,

specify requirements on when the access rule is applicable, and

contain options that can be set to be dependent on properties of the object or the subject.

98. The system of claim 77 wherein the system further comprises facilities for expressing constraints as at least:

1) relational operations on integers;

2) relational operations on strings; and

3) set operations.

99. The system of claim 77 wherein the system further comprises facilities that allow the user to define conditions.

100. The system of claim 77 wherein the system includes an Application Programming Interface (API) for invoking user-supplied code to evaluate user-defined functions.

101. A system comprising a computer having a security policy that includes at least one or more components having at least a set of privileges that includes at least one privilege that is capable of at least:

being granted to the user explicitly; and

being granted to a role which is granted to the user.

102. The system of claim 101 wherein:

the role is a named group of privileges containing at least one privilege that are granted to at least one user or to at least one other role; and

the at least one user granted to the role is a member of the role.

103. The system of claim 101 wherein the members of a role automatically inherit all the privileges granted or denied to the role.

104. The system of claim 101 wherein roles are organized into a role hierarchy, where parent roles are granted to children roles such that:

if a parent role is granted a privilege, then the children roles are automatically granted the privilege; and

if a role is denied a privilege, then the children roles are automatically denied the privilege.

105. The system of claim 101 wherein roles of an object may be defined as being local to that object.

106. The system of claim 101 wherein the role is at least a global role mapped to at least a set of local roles, having at least one role per object.

107. The system of claim 101 wherein the system further comprises more than one role, two of which have memberships that are mutually exclusive with respect to one another.

108. The security system comprising a policy manager located on a computer system that includes at least:

a management console or station;

a database management system;

an audit facility; and

a distributor.

109. The system of claim 108 wherein the management station further comprises a Graphical User Interface (GUI) for creating and customizing rules by system users.

110. The system of claim 108 wherein the management station supports concurrent rule development by multiple users.

111. The system of claim 108 wherein the management station includes an application guard to allow only authorized administrators to operate the management station based on at least a local administrative policy which provides a set of policy rules specifying which users are authorized to access management station.

112. A security system comprising:

at least one application guard stored on a computer readable medium and guards a protected application by preventing unauthorized transactional access to at least a portion of said application.

113. A security system comprising:

an application guard located within non-volatile memory that is designed to reside along with each protected application and supports transactional access control by allowing an application to detect an authorization service and to make authorization requests at each user interaction, data request, and business-level transaction.

114. The security system of claim 113 further comprising a distributor capable of distributing the application guard on clients throughout an enterprise.

115. The system of claim 113 wherein the application guard is integrated into the application through an application programming interface (API) or authorization library that allows the application to request authorization services as needed through an application guard interface.

116. The system of claim 113 further comprising

an authorization engine that processes an authorization request;.

a checker that parses local client policy and stores the parsed local client policy in Random Access Memory (RAM); and

an evaluator that determines whether the authorization request should be granted or denied by evaluating the authorization request with the parsed local client policy in RAM.

117. The system of claim 113 wherein the authorization engine comprises plug-ins that at least allow for additional capabilities to process and evaluate authorization requests based on customized code.

118. The system of claim 113 further comprising a logger where at least:

each authorization request is then recorded in an audit log; and

each authorization request made at a location remote from the logger is transmitted to the logger via a communication interface.

119. The system of claim 113 wherein the system is capable of implementing at least:

the application guard locally to the application; and

the application guard as a remote authorization service through a remote procedure call to another server.

120. A security system comprising:

at least one application guard stored on a computer readable non-volatile memory medium that is designed to reside along with each protected application and

guards a protected application by preventing unauthorized transactional access, and

supports transactional access control by allowing an application to detect an authorization service and to make authorization requests at each user interaction, data request, and business-level transaction;

wherein

the application guard is integrated into the application through an application programming interface (API) or authorization library that allows the application to request authorization services as needed through an application guard interface, and

the system is capable of implementing the application guard locally to the application and is capable of implementing the application guard as a remote authorization service through a remote procedure call to another server.

121. A computer readable storage medium having stored thereon a method for maintaining security in a distributed computing environment comprising the steps of:

managing a security policy via a policy manager; and

managing access via an application guard to a transaction related with an

application as specified by the security policy.

122. A method for maintaining security in a distributed computing environment, comprising:

managing a security policy via a policy manager; and

managing access via an application guard to a transaction referenced by an application as specified by the security policy.

123. A method of using a security system comprising:

using a management station to create or modify a policy rule

distributing the policy rule to appropriate clients via a communication interface included in the management station.

124. The method of claim 123 further comprises reviewing and reconstructing the policy rules via a parser to make sure that the policy rules are syntactically and semantically correct according to a predefined policy language.

125. The method of claim 123 further comprises determining via a differ-program the changes were made to optimized the policy, and wherein the step of distributing then distributes only the changed policy rules or local client policy to the appropriate application guards, which enforce access control to local applications and data.

126. The method of claim 123 wherein each application guard has its own specific local client policy.

127. The method of claim 123 further comprising optimizing via the distributor an administrative policy into an optimized administrative policy or local administrative policy for use with an application guard in the management station.

128. The method of distributing at least one security policy rule comprising:

passing the policy rule through at least

a DataBase layer (DB layer) and

an Open DataBase Connectivity layer (ODBC);

and

storing the policy rule as an enterprise policy.

129. The method of claim 128, wherein:

the DB layer formats the policy rules into standard database storage tables, and

the ODBC provides a common interface to various vendor-specific databases.

130. The method of claim 128 wherein the distribution occurs through the ODBC layer and a communication interface.

131. The method of claim 128 further comprising passing the enterprise policy to a distributor.

132. The method of claim 128 further comprising determining via an optimizer program within the distributor which application guard needs to receive the policy rules.

133. A method of configuring a security system comprising:

installing a policy manager on a server including

installing

a management station,

a distributor,

a logger, and

a DataBase Management System (DBMS);

entering a set of policy rules

installing application guards and local client policies onto client systems; and

registering plug-ins into the application guards to allow for additional capabilities in order to process authorization requests based on customized code.

134. The method of claim 133 wherein the step of entering includes presenting an administrator with the choice of whether to use a policy loader or management station to enter the policy rules.

135. The method of claim 133 wherein

if the administrator chooses to use the management station, then the step of entering includes using an edit function to enter the policy rules, and

if the administrator chooses the policy loader, then the step of entering

includes entering the policy rules into a file, and

passing the file to the policy loader.

136. A method of managing policy under management services in a management station comprising:

an authorized administrator logging into a policy manager;

the authorized administrator chooses between administrative mode to manage administrative policy or enterprise mode to manage enterprise policy;

presenting the administrator with menu options including

navigate tree,

analyze policy,

edit policy,

distribute policy,

view audit log which is a security feature that allows an administrator to view and track authorization requests have occurred at an application guard connected to a system, and

exit.

137. The method of claim 136 wherein menu option navigate tree provides a set of edit options for an administrator that include to

add,

delete, and

modify features;

wherein the administrator is presented with a choice of features on a server and on a client.

138. The method of claim 136 wherein the features that the administrator has the option apply the edit options include

global users,

global roles,

directories,

local roles,

local users,

applications,

application guards, and

declarations.

139. The method of claim 136 wherein the menu option analyze policy allows an authorized user to analyze and view rules and policies within enterprise policy.

140. The method of claim 136 wherein the user is presented with an option to

search rules, and

to query policy.

141. The method of claim 140 wherein if search rules is selected, the administrator is presented with options of searching grant rules and all the deny rules pertaining to a particular user.

142. The method of claim 140 wherein if query policy is selected, a search can be made on who is granted or denied what privilege on which objects under what conditions.

143. The method of claim 140 wherein the menu option edit policy presents an authorized user with the option to add, delete, and modify enterprise policy features.

144. The method of claim 143 wherein the features that may be edited include

a rule set,

access,

a privilege,

an objects,

an user

a role, and

an attribute.

145. The method of claim 136 wherein the menu option distribute policy includes distributing the new features of a newly entered or modified enterprise policy to appropriate application guards.

146. The method of claim 136 wherein upon selecting the distribute policy option

a distributor optimizes enterprise policy;

a differ program computes a difference between a newly optimized policy and a former optimized policy;

the newly optimized policy is then published as optimized policy in DBMS;

committing only the changed portions of the newly optimized policy to an appropriate application guard;

the application guard receives the newly optimized policy;

the application guard merges the newly optimized policy into local client policy; and

the local client policy is activated to work with the application guard.

147. A method of granting client access authorization comprising:

using an application guard that includes at least

requesting access to a software securable component associated with an application protected by an application guard, wherein the application guard constructs and issues an authorization request, and

evaluating the authorization request via the application guard according to its local client policy to determine whether to allow or deny the authorization request; and

an audit records the authorization request in an audit log;

wherein

if there is an error in the authorization request, or if the request is not valid, then the user is denied access;

if the authorization request is valid, then a determination is made whether access should be granted, and

if the evaluated authorization request does not deny access for the user, then access is allowed, and

if the evaluated authorization request denies access for the user, then access is denied.

148. The method of claim 147 wherein evaluating the authorization request includes an evaluator searching deny rules in local policy

if the evaluator finds a deny rule, then an evaluation is performed on any constraints on the deny rule

if the evaluation finds a presently valid constraint on the deny rule, then access is denied, and

if the evaluation finds that all constraints on the deny rule are not presently valid, then a search for a grant rule is performed;

and

if no deny rules are found, then a search for a grant rule is performed;

wherein after a search for a grant rule

if no grant rule is found that would allow access for the user, then access is denied, and

if a grant rule is found, then an evaluation is performed on any constraints in the grant rules wherein

if the evaluated constraint is presently valid, then access is allowed, and

if the evaluated constraint is not presently valid, then access is denied.

149. A method for securing a computer system comprising:

guarding a protected application by using an application guard to prevent unauthorized transactional access.

150. A method for providing a security system comprising:

providing at least one application guard that is storable on a computer readable medium and guards a protected application by preventing unauthorized transactional access to at least a component associated with the application.

151. A method for updating a security system comprising:

updating a set of policy rules containing at least one policy rule in a central location;

generating changes to the set of policy rules resulting from the updating step; and

distributing the changes to the set of policy rules.

152. The method of claim 151 wherein the policy rule contains entitlement information related to at least one resource.

153. The method of claim 151 wherein the policy rules are stored in a database table.

154. A method for establishing a security system comprising:

establishing a set of policy rules containing at least one policy rule in a central location; and

distributing the set of policy rules for enforcement.

155. The method of claim 154 wherein the policy rule contains entitlement information related to at least one resource.

156. The method of claim 154 wherein the policy rules are stored in a database table.